



15W AF

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200209059-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Jay D. Knitter

Confirmation No.: 9770

Application No.: 10/635,815

Examiner: Christopher D. Biagini

Filing Date: 08/07/2003

Group Art Unit: 2142

Title: METHOD AND APPARATUS FOR IDENTIFYING A MESSAGE SOURCE IN A NETWORK

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on January 24, 2008.

- ☐ The fee for filing this Appeal Brief is \$510.00 (37 CFR 41.20).
☒ No Additional Fee Required.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

- ☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month
\$120

☐ 2nd Month
\$460

☐ 3rd Month
\$1050

☐ 4th Month
\$1640

- ☐ The extension fee has already been filed in this application.

- ☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of \$ 00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,

Jay D. Knitter

By William T. Ellis Reg. No. 57,396

William T. Ellis

Attorney/Agent for Applicant(s)

Reg No. : 26,874

Date : April 17, 2008

Telephone : (202) 672-5485



Atty. Dkt. No. 200209059-1

***IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES***

Appellant: Jay D. Knitter
Title: METHOD AND APPARATUS FOR IDENTIFYING
A MESSAGE SOURCE IN A NETWORK
Appl. No.: 10/635,815
Filing Date: 08/07/2003
Examiner: Christopher D. Biagini
Art Unit: 2142
Confirmation Number: 9770

**RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF AND
RESUBMISSION OF A REVISED BRIEF ON APPEAL**

Mail Stop Appeal Brief - Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Response is responsive to the Notification of Non-Compliant Appeal Brief dated April 1, 2008, concerning the above-identified application. The Response is being timely filed and no fee is believed to be due.

Paragraph two of the Notification indicates that the brief does not identify the appealed claims. However, the first sentence in the "Status of Claims" section of the Brief on Appeal sets forth that the present appeal is directed to claims 1-22. Appellant's representative contacted Patent Appeal Specialist, Cassandra Paris, via telephone on Wednesday April 9, 2008 to discuss what correction, if any, was required. Mrs. Paris acknowledged that this reason was given in error and that no further correction was required under paragraph two.

Paragraph ten states that the first heading in the arguments section is incorrect. In response, Appellant has amended the first heading under the arguments section to read "Rejection of Claims 1-22 under 35 U.S.C. § 112, First Paragraph."

Accordingly, applicants believe that the Brief on Appeal, submitted on March 24, 2008, is now fully compliant with the requirements of 37 C.F.R. § 41.37. Accordingly, this Notification of Non-Compliant Appeal Brief should be satisfied.

In view of above, Appellants respectfully solicit the Honorable Board of Patent Appeals and Interferences to reverse the rejection of the pending claims and pass this application on to allowance.

REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Company.

RELATED APPEALS AND INTERFERENCES

Application No. 10/448646 incorporated by reference in the present application was filed on May 30, 2003. A Notice of Appeal was filed on July 23, 2007. An appeal brief was filed on September 20, 2007. The application has been forwarded to the Board of Patent Appeals and Interferences for a decision on the appeal.

STATUS OF CLAIMS

The present appeal is directed to claims 1-22 which are the claims under consideration. A copy of pending claims 1-22 is attached herein in the Claims Appendix.

Claims 1-22 were rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement.

Claims 6-9, 15-18 and 21 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite.

STATUS OF AMENDMENTS

Claims 1-22 were initially pending in the application filed on August 7, 2003.

Claims 1, 11, 19 and 22 were amended in an Amendment and Reply filed July 16, 2007.

Claims 1, 11, 19 and 22 were amended in an Amendment and Reply filed November 26, 2007. However, in an Advisory Action mailed December 19, 2007, the Examiner indicated that the amendments of November 26th would not be entered for the purposes of appeal.

This Appeal Brief is being filed within the statutory two month period after the filing of the Notice of appeal on January 24, 2008.

SUMMARY OF CLAIMED SUBJECT MATTER

Claims 1, 11, 19 and 22 are independent claims.

Independent claim 1 is directed to a method of identifying a message source in a network. Figure 2, step 220 and page 3, line 29 – page 4, line 5 of the application describe receiving a method call from a client computer to invoke an object on a server. Figure 2, step 230 and page 4, lines 5-9 of the application describe packaging the method call in a message to be sent from a client server to the data server via the network. Figure 2, step 240 and page 4, lines 25-28 of the application disclose identifying, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server. Figure 2, step 250 and page 4, line 28 – page 5, line 2 of the application describe transmitting the message to the data server.

Independent claim 11 is directed to a client server configured to transmit messages to a data server via a network. Figure 1, page 3, lines 17-23, Figure 2, step 220 and page 3, line 29 – page 4, line 5 of the application describe a client computer interface configured to receive a method call from a client computer to invoke an object on the data server. Figure 2, step 230 and page 4, lines 5-9 describe a data processing unit coupled to the client computer configured to package the method call in a message to be sent from a client server to the data server via the network. Figure 2, step 240 and page 4, lines 25-28 of the application disclose a data processing unit coupled to the client computer configured to identify, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server. Figure 2, step 250 and page 4, line 28 – page 5, line 2 of the application describe a data processing unit coupled to the client computer configured to transmit the message to the data server.

Claim 19 is directed to a program product, embodied in a computer readable medium, comprising machine-readable program code for causing, when executed, a computer to graphically emulate a network including at least a client computer, a client server and a data server. Figure 2, step 220 and page 3, line 29 – page 4, line 5 of the application describe the program product performing the method step of receiving a method call from a client computer to invoke an object on a server. Figure 2, step 230 and page 4, lines 5-9 of the application describe the program product performing the method step of packaging the method call in a message to be sent from a client server to the data server via the network. Figure 2, step 240 and page 4, lines 25-28 of the application disclose the program product performing the method step of identifying, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server. Figure 2, step 250 and page 4, line 28 – page 5, line 2 of the application describe the program product performing the method step of transmitting the message to the data server.

Claim 22 is directed to an apparatus configured to identify a message source in a network. Figure 2, step 220 and page 3, line 29 – page 4, line 5 of the application describe a means for receiving a method call from a client computer to invoke an object on a server. Figure 2, step 230 and page 4, lines 5-9 of the application describe a means for packaging the method call in a message to be sent from a client server to the data server via the network. Figure 2, step 240 and page 4, lines 25-28 of the application disclose a means for identifying, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server. Figure 2, step 250 and page 4, line 28 – page 5, line 2 of the application describe a means for transmitting the message to the data server.

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

Accordingly, the issue on appeal is whether the examiner erred in:

Finally rejecting claims 1-22 under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement.

Finally rejecting claims 6-9, 15-18 and 21 under 35 U.S.C. § 112, second paragraph, as being indefinite.

ARGUMENT

Rejection of Claims 1-22 under 35 U.S.C. § 112, First Paragraph

As a preliminary matter, Appellant notes that the level of predictability in the computer arts is generally considered predictable. The amount of guidance or direction needed to enable the invention is inversely related to the amount of knowledge in the state of the art as well as the predictability in the art. *In re Fisher*, 427 F.2d 833, 839, 166 USPQ 18, 24 (CCPA 1970). The more that is known in the prior art about the nature of the invention, how to make, and how to use the invention, and the more predictable the art is, the less information needs to be explicitly stated in the specification. *See* MPEP § 2164.03. Here, in contrast to the above teachings, the Examiner, while acknowledging the level of predictability in the art is high, requires that the specification provide significant guidance and direction.

As long as the specification discloses at least one method for making and using the claimed invention that bears a reasonable correlation to the entire scope of the claim, then the enablement requirement of 35 U.S.C. § 112 is satisfied. *In re Fisher*, 427 F.2d 833, 839, 166 USPQ 18, 24 (CCPA 1970). Paragraphs [0015]-[0022] and Figs. 2-5 of the present application disclose a method for carrying out the claimed invention. Further, in view of the fact that significant guidance in the art is not required and Application Nos. 10/448,646 and 10/449,555 are incorporated by reference, Appellant respectfully traverses the rejection and requests that the rejection be withdrawn.

In the Office Action dated September 26, 2007, the Examiner sets forth several factors in support of the assertion that claims 1-22 fail to comply with the enablement requirement. In the Office Action, the Examiner argues that the “features that Appellant argues distinguish the claims from the prior art are those that are not adequately described.” Specifically, the Examiner asserts that the Appellant argued that the distinguishing feature of the claimed invention is the use of a comparison algorithm on a client server to identify an object on the client computer that is invoking the object on [a] data server but the specification does not adequately describe how this operation occurs. Appellant respectfully disagrees.

Claims 1-22 are enabled by the specification, which includes U.S. Patent Application Nos. 10/448,646 and 10/449,555 which were incorporated by reference on page 1 of the specification. The claimed method recites that an object on a client computer is identified

from an execution stack using a comparison algorithm. On page 6 of the September 26th Office Action, the Examiner argues that the specification does not adequately describe how a comparison algorithm on a client server is used to identify an object on the client computer that is invoking the object on a data server. However, paragraph [0019] of the specification clearly sets forth this operation. Paragraph [0019] states “[t]he identifier (fully qualified class name) of the source of the SOAP call is stored in a SOAP header which is part of the message transmitted from the client computer 11 to the data server 140 (via client server 120).” The identifier indicates the object on the client computer that is invoking the object on the data server. As disclosed in paragraph [0019] the identifier is transmitted in the header of the method call that is transmitted from the client computer to a data server via the client server. The identifier is a portion of the execution stack and identifies the object name that invoked the method to send the message to the data server. *See* page 7, lines 27-29 (“However, the entire stack is not in the header, only the class name that invoked the method to send the message to the data server 140.”). Since a portion of the execution stack of the client computer, namely an identifier indicating the object on the client computer that is invoking the object on the data server is included in the header of the method call sent from the client computer to the client server, the client server is able to examine the header of the method call to identify the object that is invoking the object on the data server. *See* page 7, lines 29-30 (“The identifier can the [sic] be retrieved by an administrator by simply examining the header of the message.”).

Paragraph [0016] specifically recites an algorithm for implementing the claimed invention. The code in paragraph [0016] is for illustration purposes only. The client class generates a method call (sendSOAPMessage()) to invoke an object on the data server. The method sendSOAPMessage() creates a new class Client3 which invokes a method called findSourceOfSOAPMessage(). The findSourceOfSOAPMessage() method uses an algorithm (shown on page 6) to find and return the source of the SOAP message (object that generates the method call). This code is executed on the client computer. The source of the SOAP message is identified in a SOAP header which is apart of the method transmitted to the data server via client server. As stated above, since a portion of the execution stack of the client computer, namely an identifier indicating the object on the client computer that is invoking

the object on the data server is included in the header of the method call sent from the client computer to the client server, the client server is able to examine the header of the method.

In the Final Office Action, the Examiner states that the client executes on the same machine as the comparison algorithm and would not be functional in the claimed embodiment. However, Appellant notes that a “client server” need not be a separate machine. The client server can be a program that runs on the client computer. *See* Corba Glossary (<http://www.ooportal.com/corba-fundamentals/glossary.html>), last viewed March 10, 2008. Accordingly, the claims, given their most reasonably broad interpretation, would cover a system where the client server is a program running on the client computer. Further, one skilled in the art, e.g., Java programming, given the algorithm in paragraph [0016] would be able to implement the algorithm in a client server environment (where the client and server are separate machines) using any combination of Java technologies including, for example, Java Server Pages, Java Servlets, Java Enterprise Beans and similar technologies. Moreover, in view the fact that less information needs to be explicitly stated in the specification because the level of predictability is high, the totality of evidence suggests that it would not require undue experimentation to make and use the claimed invention.

Rejection of Claims 6-9, 15-18 and 21 under 35 U.S.C. § 112, Second Paragraph

Claims 6-9, 15-18, and 21 were rejected as being indefinite for failing to point out which Simple Object Access Protocol (SOAP) the claims are directed to. As stated in the “Background of the Invention,” SOAP is a “protocol layered on top of HTTP ... which allows Automation objects to be invoked over the Internet via Web servers.” Further, a precise and clear definition of the SOAP protocol can be obtained from paragraph [0004] of the specification which describes briefly how the SOAP protocol operates.

Given the definition set forth in the specification and the clear context given in paragraph [0004], the term SOAP has a clear meaning which is sufficiently precise and definite.

The protocol described in U.S. Patent No. 6,457,066 (“the ‘066 patent”) and the protocol implemented by Apache Axis were included as examples of implementations of SOAP. In other words, these implementations are only subsets of a larger SOAP class that is compatible with the applicant’s invention. Thus claims 6-9, 15-18, and 21 do not refer only

to the '066 patent's SOAP implementation or to the Apache Axis SOAP implementation, but the claims refer to any implementation that a person having ordinary skill in the art would recognize as a SOAP implementation. The wide ranging application of the applicant's invention is buttressed by its compatibility with "other software and protocols, such as various TCP messaging protocols." ([0011]).

Accordingly, Applicant respectfully requests that the rejection of claims 6-9, 15-18 and 21 be withdrawn.

CONCLUSION

In view of above, appellants respectfully solicit the Honorable Board of Patent Appeals and Interferences to reverse the rejections of claims 1-22 under 35 U.S.C. § 112 and pass this application on to allowance.

At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 C.F.R. § 1.25. Additionally, charge any fees to Deposit Account 08-2025 under 37 C.F.R. § 1.16 through § 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,

Date 4/17/08

HEWLETT-PACKARD COMPANY
Customer Number: 22879
Telephone: (202) 672-5485
Facsimile: (202) 672-5399

By W. T. Ellis

William T. Ellis
Registration No. 26,874
Walter K. Robinson
Registration No. 59,396
Attorneys for Appellant

CLAIMS APPENDIX

1. (Previously Presented) A method of identifying a message source in a network, comprising:

receiving a method call from a client computer to invoke an object on a data server;

packaging the method call in a message to be sent from a client server to the data server via the network;

on the client server, identifying, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server; and

transmitting the message to the data server.

2. (Original) The method of claim 1, further comprising:

on the client computer, generating the method call to invoke the object on the data server.

3. (Original) The method of claim 2, wherein transmitting the message to the data server transmits an identifier of an object on the client computer invoking the object on the data server along with the message.

4. (Original) The method of claim 3, wherein the identifier is stored in a header of the message.

5. (Original) The method of claim 3, wherein the identifier comprises a fully qualified class name.
6. (Original) The method of claim 1, wherein the message comprises a simple object access protocol (SOAP) message.
7. (Original) The method of claim 6, wherein packaging the method call in a message comprises building up a SOAP request.
8. (Original) The method of claim 7, wherein transmitting the message comprises implementing a SOAP application programming interface (API).
9. (Original) The method of claim 8, wherein the SOAP API comprises a messaging API.
10. (Original) The method of claim 2, further comprising:

displaying a Web service graphical component representing the object; and

displaying an interconnecting graphical component representing an associated interaction between the client computer and the data server.
11. (Previously Presented) A client server configured to transmit messages to a data server via a network, comprising:

a client computer interface configured to receive a method call from a client computer to invoke an object on the data server; and

a data processing unit coupled to the client computer interface, the data processing unit being configured to:

package the method call in a message to be sent from the client server to the data server via the network;

identify, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server; and

transmit the message to the data server.

12. (Original) The client server of claim 11, wherein the message is transmitted along with an identifier of an object on the client computer invoking the object on the data server.
13. (Original) The client server of claim 12, wherein the identifier is stored in a header of the message.
14. (Original) The client server of claim 12, wherein the identifier comprises a fully qualified class name.
15. (Original) The client server of claim 11, wherein the message comprises a simple object access protocol (SOAP) message.

16. (Original) The client server of claim 15, wherein packaging the method call in a message comprises building up a SOAP request.
17. (Original) The client server of claim 16, wherein transmitting the message comprises implementing a SOAP application programming interface (API).
18. (Original) The client server of claim 17, wherein the SOAP API comprises a messaging API.
19. (Previously Presented) A program product, embodied in a computer readable medium, comprising machine-readable program code for causing, when executed, a computer to graphically emulate a network including at least a client computer, a client server, and a data server, the program product graphically emulating the network performing method steps of:
 - on the client computer, generating a method call to invoke an object on the data server;
 - packaging the method call in a message to be sent from the client server to the data server via the network;
 - on the client server, identifying an identifier of an object on the client computer invoking the object on the data server from an execution stack through a comparison algorithm; and
 - transmitting the message to the data server.

20. (Original) The program product of claim 19, wherein the identifier comprises a fully qualified class name.
21. (Original) The program product of claim 19, wherein the message comprises a simple object access protocol (SOAP) message.
22. (Previously Presented) An apparatus configured to identify a message source in a network, comprising:
 - means for receiving a method call from a client computer to invoke an object on a data server;
 - means for packaging the method call in a message to be sent from a client server to the data server via the network;
 - means, on the client server, for identifying, from an execution stack and through the use of a comparison algorithm, an object on the client computer that is invoking the object on the data server; and
 - means for transmitting the message to the data server.

EVIDENCE APPENDIX

Server: A server is a program that awaits and fulfills requests from client programs in the same or other computers. *See* Corba Glossary (<http://www.ooportal.com/corba-fundamentals/glossary.html>), last viewed March 10, 2008.

RELATED PROCEEDINGS APPENDIX

None.